

04-28-00

A

ASSISTANT COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, DC 20231

PATENT
Date: April 27, 2000
File No. 0941.63996

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): Takaoki Sasaki, Takeshi Hatano, Ryota Akiyama,
Seigo Kotani, Takayuki Hasebe

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: Asst. Comm. for Patents, Washington, D.C. 20231, on this date.

04-27-00
Date

Express Mail Label No.: EL409492192US

For: FILE PROCESSING UNIT

Enclosed are:

- (X) 34 pages of specification, including 17 claims and an abstract.
- (X) an executed oath or declaration, with power of attorney.
- () an unexecuted oath or declaration, with power of attorney.
- () sheet(s) of informal drawing(s).
- (X) 10 sheet(s) of formal drawings(s).
- (X) Assignment(s) of the invention to Fujitsu Limited.
- (X) Assignment Form Cover Sheet.
- (X) A check in the amount of \$40.00 to cover the fee for recording the assignment(s) is enclosed.
- () Information Disclosure Statement.
- () Form PTO-1449 and cited references.
- () Associate power of attorney.
- (X) Priority Document.

Fee Calculation For Claims As Filed

a) Basic Fee						\$ 690.00
b) Independent Claims	<u>5</u>	-	3	=	<u>2</u>	x \$ 78.00 = \$ <u>156.00</u>
c) Total Claims	<u>17</u>	-	20	=	<u>0</u>	x \$ 18.00 = \$ <u> </u>
d) Fee for Multiple Claims						\$260.00 = \$ <u> </u>
Total Filing Fee						\$ <u>846.00</u>

- (X) A check in the amount of \$ 846.00 to cover the filing fee is enclosed.
- (X) The Commissioner is hereby authorized to charge any additional fees which may be required to this application under 37 C.F.R. §§1.16-1.17, or credit any overpayment, to Deposit Account No. 07-2069. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 07-2069. A duplicate copy of this sheet is enclosed.

Suite 8660 - Sears Tower
233 S. Wacker Drive
Chicago, Illinois 60606
(312) 993-0080

GREER, BURNS & CRAIN, LTD.

By: [Signature]
Patrick G. Burns
Registration No. 29,367

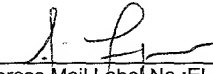
04/27/00
jc777 U.S. PTO

jc525 U.S. PTO
09/559138
04/27/00

09559138-042700

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: Asst. Comm. for Patents, Washington, D.C. 20231, on this date.

04-27-00
Date


Express Mail Label No.: EL409492192US

SPECIFICATION

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN THAT WE, Takaoki Sasaki, a citizen of Japan residing at Minato-ku, Tokyo, Japan, Takeshi Hatano, a citizen of Japan residing at Minato-ku, Tokyo, Japan, Ryota Akiyama, a citizen of Japan residing at Kawasaki-shi, Kanagawa, Japan, Seigo Kotani, a citizen of Japan residing at Kawasaki-shi, Kanagawa, Japan and Takayuki Hasebe, a citizen of Japan residing at Kawasaki-shi, Kanagawa, Japan have invented certain new and useful improvements in

FILE PROCESSING UNIT

of which the following is a specification : -

004240" 8616550

TITLE OF THE INVENTION

FILE PROCESSING UNIT

BACKGROUND OF THE INVENTION

5 1. Field of the Invention

The present invention generally relates to file processing units that perform processes such as file open, file close, read, and write, based on accesses made to an external storage device by an application, and, more particularly, to a file processing unit that produces and checks signature information for a designated file.

2. Description of the Related Art

15 In order to determine whether the data in a file produced utilizing a computer system and stored in an external storage (such as hard disk, optical disk, or floppy disk) has been altered by a third party, signature data is affixed to the file and then checked. Conventionally, such signature data has been produced and checked by means of a special application. More specifically, such signature data is produced and checked in the following manner.

25 After a file produced by an application is stored in an external storage, a special application designed for producing and checking signature data is started. The file is then read out of the external storage in accordance with the special application, and signature data is produced
30 utilizing all or a part of the data of the read out file. The signature data attached to the corresponding read out file is again stored in the external storage. When the data of the file stored in the external storage in accordance with the
35 special application is to be edited, the special application is started before the file is accessed. In accordance with the special application, the data

002240 08T65560

of the designated file is read out of the external storage, and signature data is produced utilizing all or a part of the data of the designated file. The signature data produced here is then compared
5 with the signature data already attached to the designated file. If both signature data are identical, it is determined that the data in the designated file has not been altered. If the signature data are not identical, it is determined
10 that the data in the designated file has been altered after the older signature data was produced.

Based on the signature data result, a user can determine whether the data of the designated file has been altered. After the determination, the
15 user reads the data of the designated file from the external storage, and performs processes such as editing.

In the conventional system described above, every time a file is processed by an application, it
20 is necessary to start the special application for producing and checking signature data. This makes the signature producing and checking procedure complicated and troublesome.

25 SUMMARY OF THE INVENTION

A general object of the present invention is to provide file processing units in which the above disadvantages are eliminated.

A more specific object of the present
30 invention is to provide a file processing unit which can produce or check a signature when accessing a file stored in an external storage unit through any application.

The above objects of the present invention
35 are achieved by a file processing unit comprising:
a first signature information producing unit that produces first signature information, in

09569438.042700

5 a signature information storage control
unit that correlates the first signature information
with the file designated by the close request, and
stores the first signature information in a storage
unit.

20 The above objects of the present invention
are also achieved by a file processing unit
comprising:

a signature agreement determining unit
30 that determines whether the second signature
information is identical to first signature
information produced utilizing the data of the file
designated by the open request and stored, attached
to the designated file, in a storage unit; and
35 a determination result producing unit that
produces information representing a determination
result of the signature agreement determining unit.

the information being supplied to the application.

In this file processing unit, it is not necessary to start an application especially designed for producing signature information. When
5 an open request to open a file used in any application is issued, the first signature information stored together with the file designated by the open request in the storage unit is compared with the second signature information produced
10 utilizing the data of the file designated by the open request. A result information is then produced based on the comparison result. Thus, signature information can be easily produced when a file used in any application is opened.

15 The above objects of the present invention are also achieved by a file processing unit that is applied to a computer system having a file system that performs processes on a file based on an event issued from an application, the file processing unit
20 comprising:

a first signature information producing unit that produces first signature information in accordance with a predetermined rule using the data of a file designated by a file close event issued
25 from the application, before a process specified by the file close event is completed in the file system;

a signature information storage control unit that attaches the first signature information
30 to the file designated by the file close event, and stores the first signature information in a storage unit; and

a first returning unit that performs, after the first signature information is produced, a
35 process for returning to the process for the file specified by the file close event in the file system.

When the file system in the computer

004240" 88T.65560

system performs in accordance with a file close event, this file processing unit receives the file close event, and produces the first signature information utilizing the data of the file

5 designated by the file close event. The first signature information is stored, together with the file, in the storage unit. The file processing unit then performs a return process for returning to the process for the file specified by the file close

10 event in the file system. Accordingly, while the file system and the application perform the process specified by the file close event, the first signature information is automatically produced, and is stored, together with the file designated by the

15 file close event, in the storage unit.

The above objects of the present invention are also achieved by a file processing unit that is applied to a computer system having a file system that performs processes on a file based on an event

20 issued from an application, the file processing unit comprising:

a second signature information producing unit that produces second signature information in accordance with a predetermined rule using the data

25 of a file designated by a file open event issued from the application, before a process specified by the file open event is completed in the file system;

a signature agreement determining unit that determines whether the second signature

30 information is identical to first signature information produced in accordance with the predetermined rule using the data of the file designated by the file open event and stored, together with the designated file, in a storage

35 unit;

a determination result producing unit that produces information representing a determination

004240" SET 6359

result of the signature agreement determining unit,
the information being supplied to the application;
and

5 a second returning unit that passes the
information representing the determination result to
the file system, and performs a process for
returning to the process for the file specified by
the file open event in the file system.

10 When the file system in the computer
system performs in accordance with a file close
event, this file processing unit receives the file
open event, and produces the second signature
information utilizing the data of the file
15 designated by the file open event. The second
signature information is compared with the first
signature information stored together with the file
in the storage unit. The file processing unit then
sends the comparison result to the file system, and
20 performs a return process for returning to the
process for the file specified by the file open
event in the file system. Accordingly, while the
file system and the application perform the process
specified by the file open event, the second
signature information is automatically produced, and
25 is compared with the first signature information
already produced for the file.

30 The above and other objects and features
of the present invention will become more apparent
from the following description taken in conjunction
with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

35 FIG. 1 is a block diagram showing the
hardware structure of a computer system to which a
file processing unit of the present invention is
applied;

FIG. 2 is a flowchart of a driver

002240" 88T65560

registration process carried out between a driver and a file system (OS) in the system of FIG. 1;

FIG. 3 is a flowchart of an initial process carried out between the driver and the file system in the system of FIG. 1;

FIG. 4 is a flowchart of a process of authentication and mode setting between an application and the driver in the system of FIG. 1;

FIG. 5 is a flowchart of a file open process carried out among an application, the file system, the driver, and the hard disk drive unit in the system of FIG. 1;

FIG. 6 is a flowchart of a read process carried out among an application, the file system, the driver, and the hard disk drive unit in the system of FIG. 1;

FIG. 7 is a flowchart of an open process carried out among an application, the file system, the driver, and the hard disk drive unit in the system of FIG. 1;

FIG. 8 is a flowchart of a file close process carried out among an application, the file system, the driver, and the hard disk drive unit in the system of FIG. 1;

FIG. 9 shows an example mode table used in the system of FIG. 1; and

FIG. 10 shows the relationship between folders and processing modes used in the system of FIG. 1.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following is a description of embodiments of the present invention, with reference to the accompanying drawings.

FIG. 1 shows the structure of a computer system to which a file processing unit of an embodiment of the present invention is applied.

In FIG. 1, the system comprises a CPU 10, a memory unit 12, an input unit 14, a display unit 16, a hard disk drive unit (HDD) 18, and an interface circuit 22. The units 10 to 18 and the interface circuit 22 are all connected to a bus.

The memory unit 12 stores an operation system (OS) and various applications, and the CPU 10 performs an operation in accordance with the OS stored in the memory unit 12 and runs a selected application. The input unit 14 includes a mouse and a keyboard, and is used to select an application, input data into a file opened by a selected application, set necessary modes, and input other necessary information.

The display unit 18 displays various control pages and information of a file opened by an application. The hard disk drive unit 18 drives a hard disk 20. In compliance with instructions issued from the CPU 10 based on a selected application and the OS, the hard disk drive unit 18 stores files in the hard disk 20 or reads files out of the hard disk 20.

A PC card 100 is connected to the interface circuit 22. The PC card 100 contains a driver as a file processing unit of the present invention. The interface circuit 22 interfaces the driver in the PC card 100 with an application run by the CPU 10 or a file system built in the OS.

When the PC card 100 is set to the interface circuit 22 after the above computer system is started, an initial process is carried out between a file system built in the OS and the driver in the PC card 100 in a manner shown in FIG. 2. FIG. 2 is a flowchart of a driver registration process carried out between the file system and the driver. In step S1, the file system sends a driver confirmation command to the driver. In step S2, it

is determined whether the initial process can be carried out in the driver. If the initial process cannot be carried out in the driver, the driver sends an error response to the file system. In step 5 S3, upon receipt of the error response, the driver performs a predetermined error process, in which an error message indicating that the driver cannot be started is displayed on the display unit 16.

If the initial process can be started in 10 the driver, the driver sends a normal response to the file system. In step S4, upon receipt of the normal response, the file system registers the information representing the driver in the memory unit 14. The file system then outputs a message 15 indicating that the driver has been registered. The message is displayed on the display unit 16.

After the driver in the PC card 100 has been registered with the PC card 100 being connected to the interface circuit 22, a process is carried 20 out according to the procedure shown in FIG. 3. In step S11, the file system in which the driver has been registered instructs the driver to carry out the initial process. Upon receipt of the instruction, the driver starts the initial process 25 in step S12, and sends a request to the file system to transfer an event from the application to the driver in step S13.

If the request is acceptable, the file system registers an address (hook address) to be 30 designated for transferring an event to the driver in step S14. On the other hand, if the initial process in the driver is incomplete, or if the file system fails to receive the request from the driver, the file system sends a request error response to 35 the driver in step S14.

Upon receipt of the request error response, the driver determines whether the request error

00240" GET6550

5

10

20

25

30

35

the mode values and the processes shown in FIG. 9 is stored as a mode table in advance in the memory in the driver.

After the processing mode setting and
5 folder designation, the processing modes and the folders are correlated with each other, and are stored in the form of a table shown in FIG. 10 in the memory in the driver. By defining the
10 relationship between the processing modes and the folders, the files to be carried out in each processing mode are stored in each corresponding folder.

After the processing mode setting and
15 folder designation, the application sends a process start instruction to the driver via the OS in step S25. Upon receipt of the instruction, the driver starts an operation in a processing mode determined based on an event (open event, close event, read
20 event, or write event) issued from the application in step S26.

Referring now to FIGS. 5 to 8, an example, in which the mode value is set at 15 (see FIG. 9) and a folder P (see FIG. 10) is designated, will be described.

25 When a user opens a file to process the data in the file while the application and the driver are in operation, the file open process is carried out according to the procedure shown in FIG. 5.

30 In step S31, an open function is issued from the application, and is passed to the file system (OS). The file system passes a hook address and an open event to the registered driver in step S32, and starts the file open process in step S33.
35 The open event includes: the name of a designated folder; a file name (including a path name); file attribute; the parameter of the open function; and a

004240" SET 6.5.56

In step S34, the driver receives the open event and then moves on to the subsequent steps.

In step S35, the designated folder is checked. In this step, it is determined whether the designated folder is registered as a folder that stores files to be processed by the driver, with reference to the table shown in FIG. 10. In this example, since the designated folder P is registered as a folder to be processed, the driver checks the file attribute, referring to an attribute table (not shown) in step S36. The file attribute check is carried out to determine whether a signature producing process or an encryption process can be performed on the file. Attribute values that indicate neither signature producing process nor encryption process can be performed are assigned to the system file, hidden files, and driver files.

30 The driver then outputs a request to read
the designated file to the hard disk drive unit 18
in step S39. In response to this request, the hard
disk drive unit 18 reads the data of the designated
file from the hard disk 29, and sends the data to
35 the driver in step S40.

Upon receipt of the data from the hard disk drive unit 18, the driver produces signature

30 If the existing signature data (B) and the
newly produced signature data (A) are determined not
to be identical in step S44, the driver sets an open
error in step S46, and sends the open error status
to the file system when returning the return address
35 in the file system. In this case, the file system
confirms the open error status and ends the file
open process in step S48. The file system then

passes the open error status as well as the result of the file open process to the application. As a result, the application returns to the open function, and the file is opened together with a message
5 indicating that the data in the file might have been altered.

In the above process based on the open event, if the designated folder is not to be checked for its signature when a file is opened, or if the
10 designated file has an attribute that does not need a signature, the driver performs the return process without carrying out the signature check in the S47.

As described above, when the open function is issued from the application to the file system,
15 the driver automatically carries out the signature check, and sends the check result to the file system. The file system then performs the normal file open process, and returns the results of the file open process and the signature check to the application.
20 Accordingly, the user can obtain the result of a signature check on a file to be opened simply by setting a processing mode when a file is opened with a desired application in operation, without starting an application especially designed for checking
25 signatures.

The flowchart of FIG. 6 illustrates a read process to be performed on the file opened in the above described manner.

As shown in FIG. 6, in step S51, a read
30 function is issued from the application to the file system. Upon receipt of the read function, the file system transmits a read event to the driver in step S52, and starts the read process in step S53. The driver receives the read event in step S54, and
35 carries out the subsequent processes. Here, the read event includes: a file name (including a path name); a file handling flag; a read point; a read

002240" 8276350

After the folder is checked in step S55 in the same manner as in the file open process, the file attribute is checked in step S56. In the folder checking step, it is determined whether the designated folder stores files containing data to be decrypted and expanded when the file is read, referring to the tables shown in FIGS. 9 and 10. In the file attribute checking step, it is determined whether a required file handling flag is set as an attribute value of the designated file in the attribute table. In this manner, it is determined whether a read process can be performed on the designated file. Since a mode value of 15 is set to the designated folder P in this example, the designated folder is determined to contain files to be decrypted and expanded during the read process. Also, since the file handling flag has been set as a file attribute value in the file open process, it is determined that the required handling flag has already been set to the attribute of the designated file in the attribute table.

In step S58, the driver outputs a request to the hard disk drive unit 18 to read the data from the designated file. Upon receipt of the request, 35 the hard disk drive unit 18 reads the data of the designated file from the hard disk 20 through a boundary process, and passes the data to the driver

in step S59. The data has been already encrypted and compressed in accordance with a predetermined algorithm described later.

After receiving the data from the hard
5 disk drive unit 18, the driver decrypts the data in
step S60. In this decrypting step, the data is
decrypted and expanded every 8 bytes in accordance
with an algorithm corresponding to the encrypting
and compressing algorithm. Here, a fraction number
10 process is also performed on the data. Each set of
8 bytes in the read data corresponds to each
corresponding set of 8 bytes in the decrypted and
expanded data, so that the both data cannot be
distinguished from each other by different data
15 lengths.

After the decrypting step, the amount of
data corresponding to the read size is stored in the
read buffer in step S61, and the driver returns to
the return address in the file system in step S62.
20 The file system then continues the read process at
the return address. After acquiring the data stored
in the read buffer, the file system finishes the
read process in step S63. The file system then
transmits the result of the read process to the
25 application.

Upon receipt of the result, the
application returns to the read function in step S64.
The application then acquires the read data, and
displays the read data on the display unit 16.

30 If the designated folder is determined in
step S55 not to be decrypted and expanded when the
file is read, or if the required file handling flag
is not set in the attribute of the designated file
in the attribute table in step S56, or if the
35 selected processing mode is determined in step S57
not to be a processing mode for decrypting and
expanding the data when the file is read, the driver

00240"GET6360

does not perform the decrypting process, and directly returns to the return address in the file system (S47). The file system then transmits the result of the read process to the application.

5 As described above, when the read function is issued from the application to the file system, the driver automatically carries out the decrypting and expanding process on the data of the designated file to be read, and sends the decrypted and
10 expanded data to the file system. The file system then performs the normal read process, and returns the decrypted and expanded data as the result of the read process to the application. Accordingly, the user can obtain data that has been decrypted and
15 expanded to the original state while an application is in operation, without carrying out any operation to decrypt and expand the encrypted and compressed data.

 Meanwhile, a write process can be carried
20 out on the file opened in the file open process, in accordance with the procedure shown in FIG. 7.

 In step S71, a write function is issued from the application to the file system. Upon receipt of the write function, the file system
25 transmits a write event to the driver in step S72, and starts the write process in step S73. The driver then carries out the subsequent steps. Here, the write event includes: a file name (including a path name); a file handling flag; a write point;
30 write data; a write buffer; write size; and a return address in the file system.

 The designated folder is checked in step S75 in the same manner as in the file open process and the read process, and the file attribute is
35 checked in step S76. In the folder checking step, it is determined whether the designated folder stores files containing data to be encrypted and

09559138 042700

compressed when data is written in the designated file, referring to the tables shown in FIGS. 9 and 10. In the file attribute checking step, it is determined whether a required file handling flag is set as an attribute value of the designated file in the attribute table. In this manner, it is determined whether a write process can be performed on the designated file. Since a mode value of 15 is set to the designated folder P in this example, the designated folder is determined to contain files to be encrypted and compressed during the write process. Also, since the file handling flag has been set as a file attribute value in the file open process, it is determined that the required handling flag has already been set to the attribute of the designated file in the attribute table.

After the above checking steps, a mode check is carried out in step S77. In this mode checking step, it is determined whether the selected processing mode is a mode for encrypting and compressing data when data is written in the designated file. Since the mode value is 15 in this example, the selected processing mode is determined to be a mode for encrypting and compressing data when data is written in the designated.

In step S78, the driver outputs a request to the hard disk drive unit 18 to read the data from the designated file. Upon receipt of the request, the hard disk drive unit 18 reads the data of the designated file from the hard disk 20 through a boundary process, and passes the data to the driver in step S79. The data has been already encrypted and compressed in accordance with a predetermined algorithm described later.

After receiving the data from the hard disk drive unit 18, the driver decrypts the data in step S80. In this decrypting step, the data is

decrypted and expanded every 8 bytes in accordance with an algorithm corresponding to the encrypting and compressing algorithm. Here, a fraction number process is also performed on the data.

5 After the decrypting step, the write data is overwritten on the decrypted data in step S81. The write data is then encrypted in step S82. In this encrypting step, the write data is encrypted and compressed every 8 bytes, and a fraction number
10 process is performed in accordance with a predetermined algorithm. Each set of 8 bytes in the encrypted and compressed write data corresponds to each corresponding set of 8 bytes in the previously encrypted and expanded data, so that the both data
15 cannot be distinguished from each other by different data lengths.

 After the encrypting step, the driver outputs a write request to the hard disk drive unit 18 in step S83. In response to the request, the
20 hard disk drive unit 18 writes the encrypted and compressed data in the designated file in the hard disk 20 through a boundary process, and sends a response to the driver in step S83.

 After the encrypted and compressed data is
25 written, the driver returns to the return address in the file system in step S86. The file system finishes the write process in step S85, and after the returning step (S86), passes the result of the write process to the application.

30 Upon receipt of the result, the application returns to the write function in step S87.

 If the designated folder is determined in S75 not to be encrypted and compressed when data is
35 written in the file, or if the required file handling flag is not set in the attribute of the designated file in the attribute table in step S76,

002240" 8275560

or if the selected processing mode is determined in step S77 not to be a processing mode for encrypting and compressing the data when the file is read, the driver does not perform the encrypting process, and
5 directly returns to the return address in the file system (S87). The file system then passes the write data from the application to the hard disk drive unit 18, so that the write data is written in the designated file in the hard disk 20.

10 As described above, when the write function is issued from the application, the driver automatically carries out the encrypting and compressing process on the write data, and writes the encrypted and compressed data into the
15 designated file. The file system then sends the result of the read process to the application. Accordingly, the encrypted and compressed data is stored in the designated file with any application in operation, without the user performing an
20 operation to encrypt and compress the data.

A file close process can be carried out on the file opened in the file open process, in accordance with the procedure shown in FIG. 8.

In step S91, a close function is issued
25 from the application in accordance with an operation from the user. The close function is sent to the file system. The file system then transmits a close event to the driver in step S92, and starts the file close process in step S93. The close event
30 includes: a file name (including a path name); file attribute; and a return address in the file system.

The driver receives the close event from the file system in step S94, and carries out the subsequent steps.

35 The designated folder is checked in step S95 in the same manner as in the file open process, and the file attribute is checked in step S96. In

the folder checking step, it is determined whether the designated folder stores files containing data to be provided with a signature when the designated file is closed, referring to the tables shown in
5 FIGS. 9 and 10. In the file attribute checking step, it is determined whether a required file handling flag is set as an attribute value of the designated file in the attribute table. Since a mode value of 15 is set to the designated folder P in this example,
10 the designated folder P is determined to contain files to be provided with signature data when the designated file is closed. Also, since the file handling flag has been set as a file attribute value in the file open process, it is determined that the
15 required handling flag has already been set to the attribute of the designated file in the attribute table.

After the above checking steps, a mode check is carried out in step S97. In this mode
20 checking step, it is determined whether the selected processing mode is a mode for producing signature data when the file is closed. Since the mode value is 15 in this example, the selected processing mode is determined to be a mode for producing signature
25 data when the file is closed.

In step S98, the driver outputs a request to the hard disk drive unit 18 to read the designated file. In response to the request, the
hard disk drive unit 18 reads the data of the
30 designated file from the hard disk 20, and passes the data to the driver in step S99.

After receiving the data from the hard disk drive unit 18, the driver produces signature data (B) using all of or a part of the data in
35 accordance with the predetermined rule in step S100. The driver then outputs a request to the hard disk drive unit 18 to attach the signature data (B) to

004240" 8265560

5 After outputting the request to the hard
disk drive unit 18 to attach the signature data (B)
to the designated file, the driver erases the file
handling flag corresponding to the designated file
in the attribute table in step S103. A normal
0 status is then set in step S104. The driver passes
the normal status to the file system, and returns to
the return address in the file system in step S105.

Upon receipt of the result of the file
20 close process, the application returns to the close
function in step S107. As a result, the file
designated by the user in the application is closed.

If the selected processing mode is not a mode for producing signature data when the file is closed (i.e., when the mode value is 0, 1, 2, 3, 8, 9, 10, or 11), the driver sets the normal status in step S104, and returns to the return process in the file system in step S105.

As described above, when the close

Accordingly, the designated file provided with the signature data can be closed with any application in operation, without the user taking the trouble of producing the signature.

For example, where decrypted and
20 compressed file data is supplied from a medium such
as a CD-ROM, the mode value 1 may be set. In this
case, when a read event is issued after the file is
opened, the encrypted and compressed file data is
automatically decrypted and expanded, so that the
25 user can use the file data.

For yet another example, where signature
35 data attached to a file is stored in a medium such
as a floppy disk to be supplied to another system,
the mode value 3 may be set. In this case,

5 For further example, where a file having
signature data is supplied by a medium such as a CD-
ROM, the mode value 4 may be set. In this case,
when a file open event is issued, signature data is
automatically produced based on the data in the file
0 to be opened, and is compared with the signature
data already affixed to the file to be opened. If
both signature data are identical, the user is
allowed to use the data in the file.

Although the storage unit for storing files is the hard disk 20 in the foregoing description, it is possible to use another storage device suitable for a computer system. In a processing mode for only either checking signature data or decrypting and expanding data (the mode value is 1, 8, or 9), a read-only recording medium, such as a CD-ROM, can be used as a storage unit.

The present invention is not limited to the specifically disclosed embodiments, but

variations and modifications may be made without departing from the scope of the present invention.

The present application is based on
Japanese priority application No. 11-125010, filed
5 on April 30, 1999, the entire contents of which are
hereby incorporated by reference.

004240" 0265560

WHAT IS CLAIMED IS:

10

20

30

35

2. The file processing unit as claimed in claim 1, wherein the first signature information producing unit has a close-file reading unit that reads the data of the designated file from the storage device, so that the first signature information is produced using the data read out by the close-file reading unit.

3. The file processing unit as claimed in claim 1, wherein the signature information storage control unit has a signature information attaching unit that attaches the first signature information to the file designated by the close request, so that

5

10

15

25

30

35

6. The file processing unit as claimed in claim 1, further comprising:

5 a second signature information producing unit that produces second signature information in response to an open request to open the file to be used in the application, using the data of the file in accordance with the predetermined rule;

10 a signature agreement determining unit that determines whether the first signature information attached to the file designated by the open request is identical to the second signature information; and

15 a determination result producing unit that produces information representing a determination result of the signature agreement determining unit, the information being supplied to the application.

20

7. The file processing unit as claimed in claim 6, wherein the second signature information producing unit has an open-file reading unit that reads the data of the file designated by the open request from the storage unit before a process requested by the open request is completed, so that the second signature information is produced using the data read out by the open-file reading unit.

30

8. The file processing unit as claimed in claim 6, further comprising a signature check determination unit that determines whether the first signature information attached to the file designated by the open request and stored in the

002240-0423350

storage unit should be checked when the open request is issued,

wherein, when the signature check determining unit determines that the first signature information attached to the file designated by the open request and stored in the storage unit should be checked, the second signature information producing unit produces the second signature information to be compared with the first signature information.

9. The file processing unit as claimed in claim 8, wherein the signature check determining unit comprises:

a mode setting unit that sets a processing mode for checking at least the first signature information attached to the file designated by an open request before the file is used in the application; and

a mode determining unit that determines whether the first signature information attached to the file designated by the open request and stored in the storage unit should be checked, in accordance with the processing mode preset by the mode setting unit.

30

~~10.~~ A file processing unit comprising:
a second signature information producing unit that produces second signature information, in response to an open request to open a file, using the data of the file in accordance with a

5

10

15

20

25

30

35

by the file close event is completed in the file system;

5 a signature information storage control unit that attaches the first signature information to the file designated by the file close event, and stores the first signature information in a storage unit; and

10 a first returning unit that performs, after the first signature information is produced, a process for returning to the process for the file specified by the file close event in the file system.

15 15. A file processing unit that is applied to a computer system having a file system that performs processes on a file based on an event issued from an application, said file processing unit comprising:

20 a second signature information producing unit that produces second signature information in accordance with a predetermined rule using the data of the file designated by a file open event issued from the application, before the process specified by the file open event is completed in the file system;

30 a signature agreement determining unit that determines whether the second signature information is identical to first signature information produced in accordance with the predetermined rule using the data of the file designated by the file open event and stored, together with the designated file, in a storage unit;

35 a determination result producing unit that produces information representing a determination

0950438 042700

result of the signature agreement determining unit,
the information being supplied to the application;
and

5 a second returning unit that passes the
information representing the determination result to
the file system, and performs a process for
returning to the process for the file specified by
the file open event in the file system.

10

09569138-042700
16. A file processing unit comprising:
a first signature information producing
15 unit that produces first signature information, in
response to a close request to close a file, using
the data of the file in accordance with a
predetermined rule, the file being used in an
application that is run in a computer system; and
20 a signature information storage control
unit that correlates the first signature information
with the file designated by the close request, and
stores the first signature information in a storage
unit,
25 the file processing unit functioning as an
external unit connected to the computer system.

30

17. The file processing unit as claimed
in claim 16, wherein the external unit is
constituted by a PC card.

FIG.1

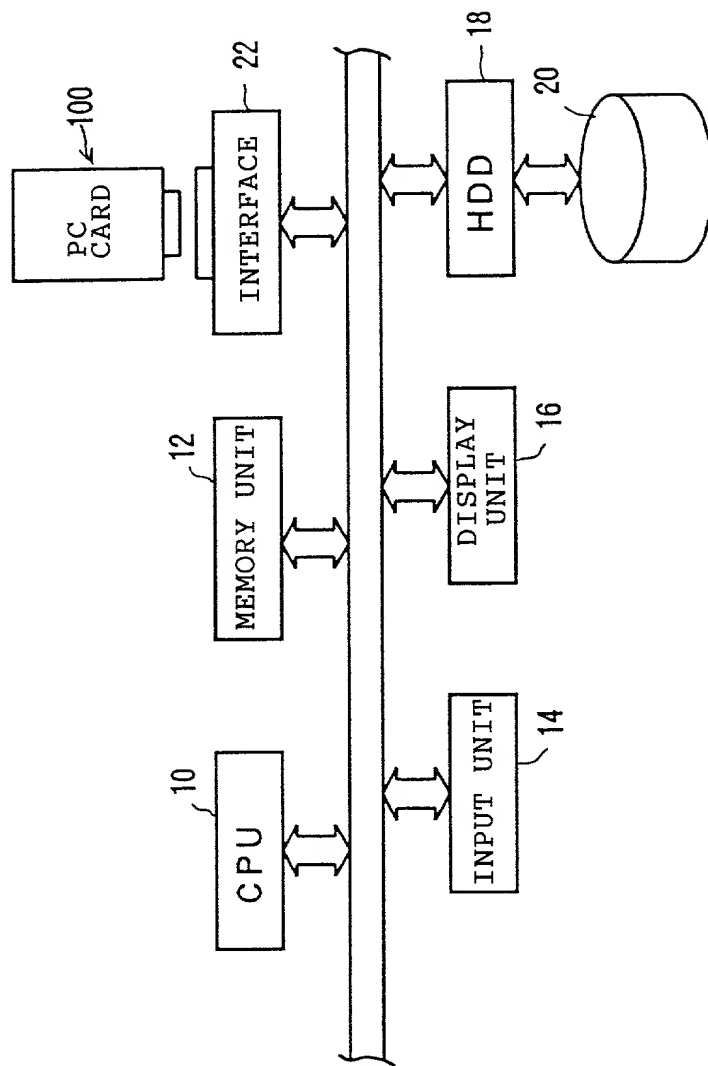


FIG.2

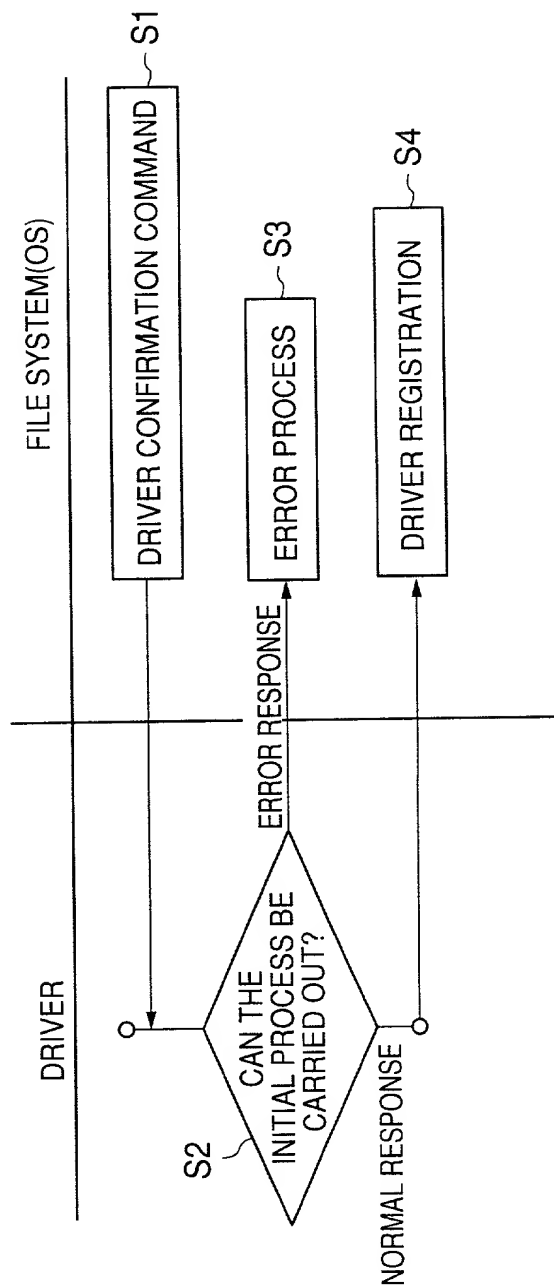


FIG.3

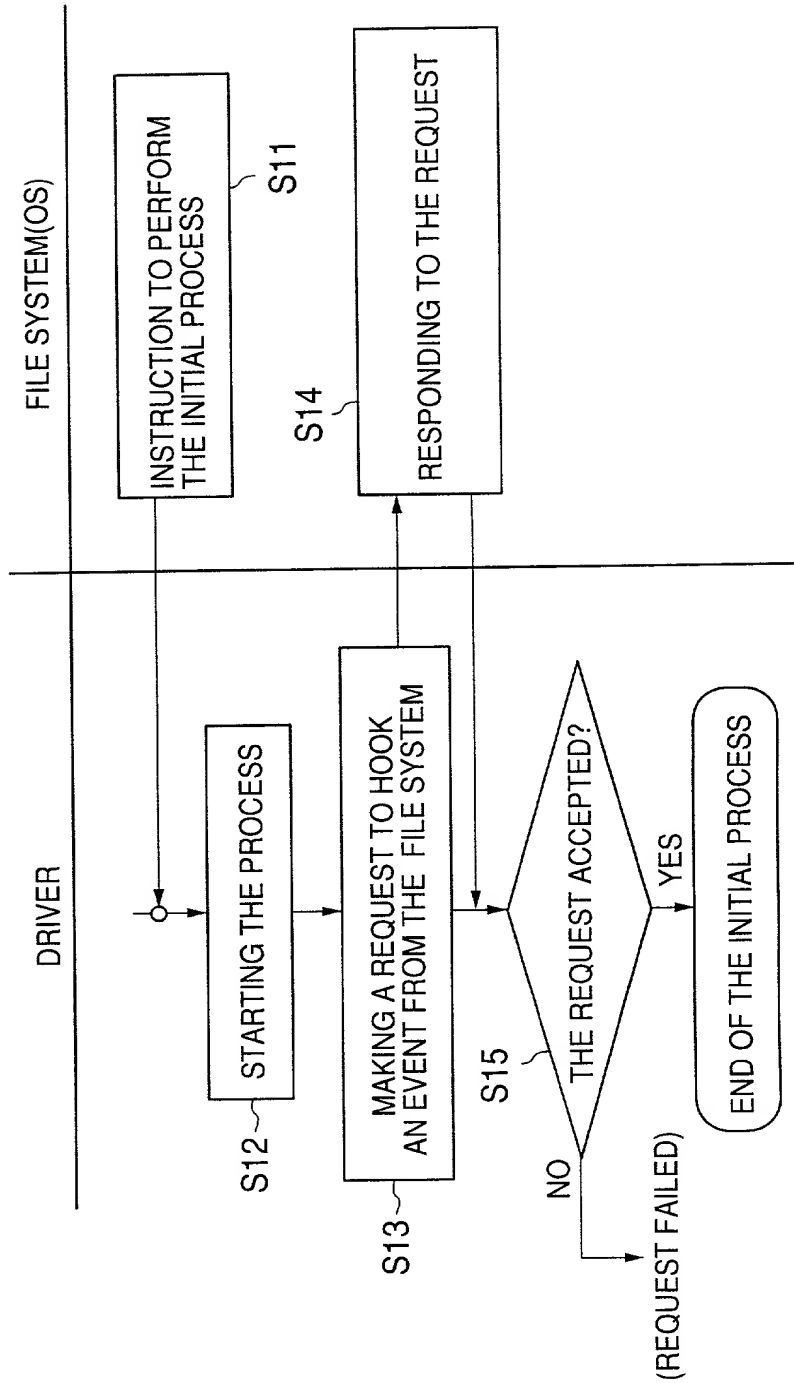


FIG.4

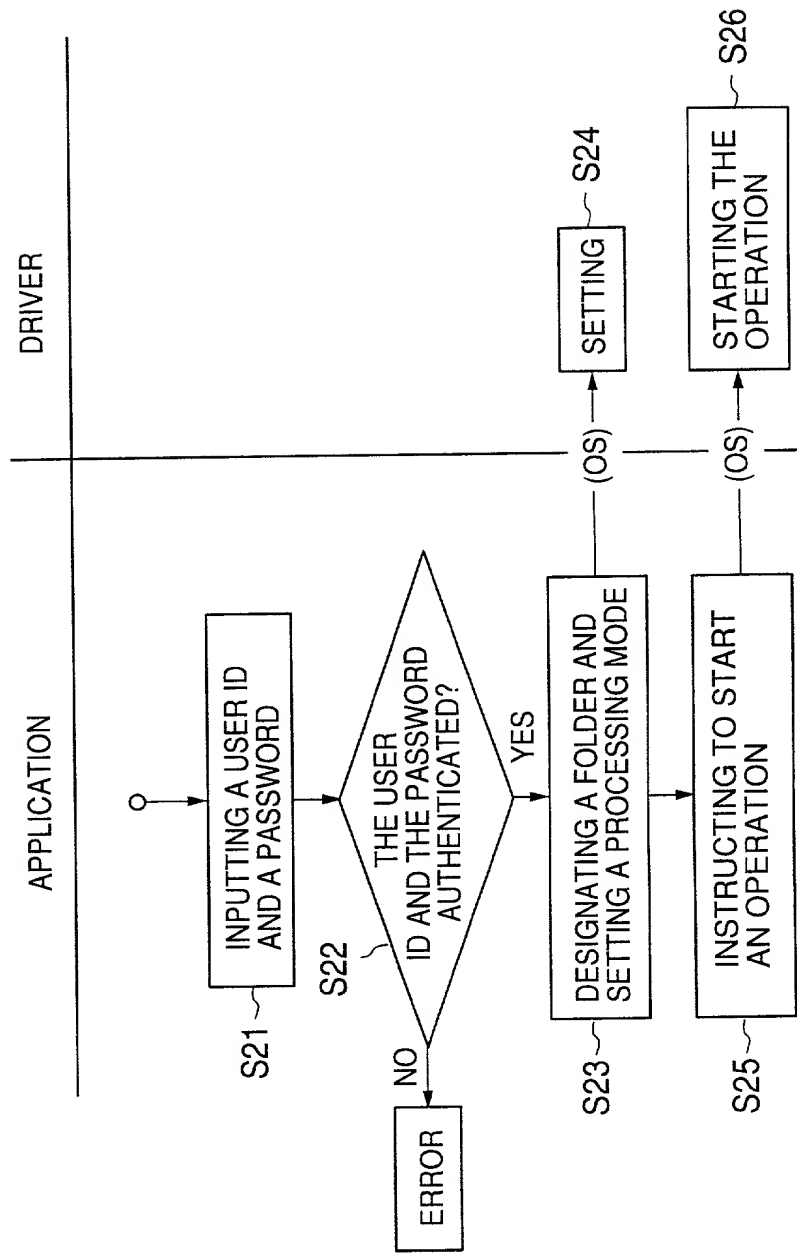
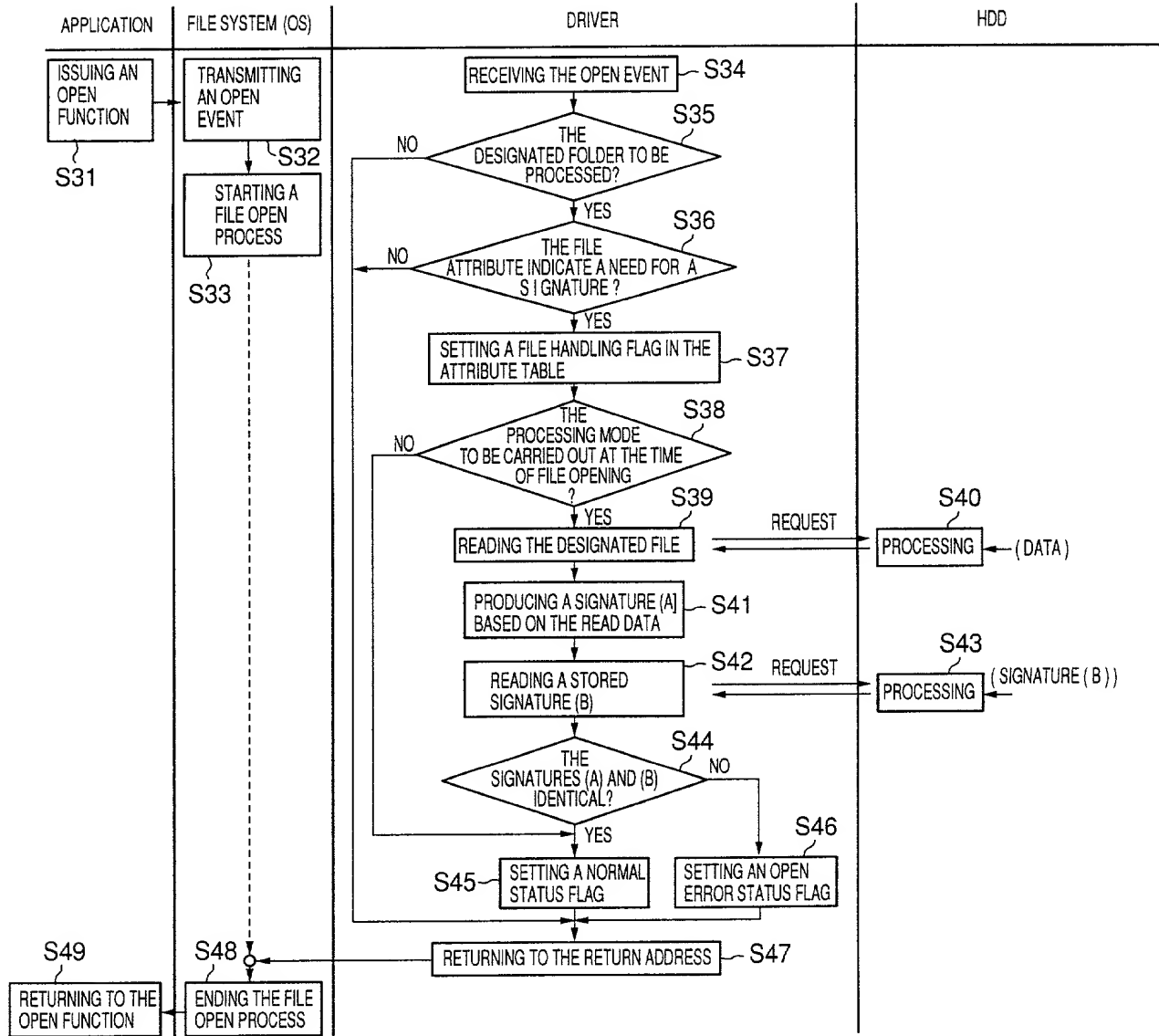


FIG.5



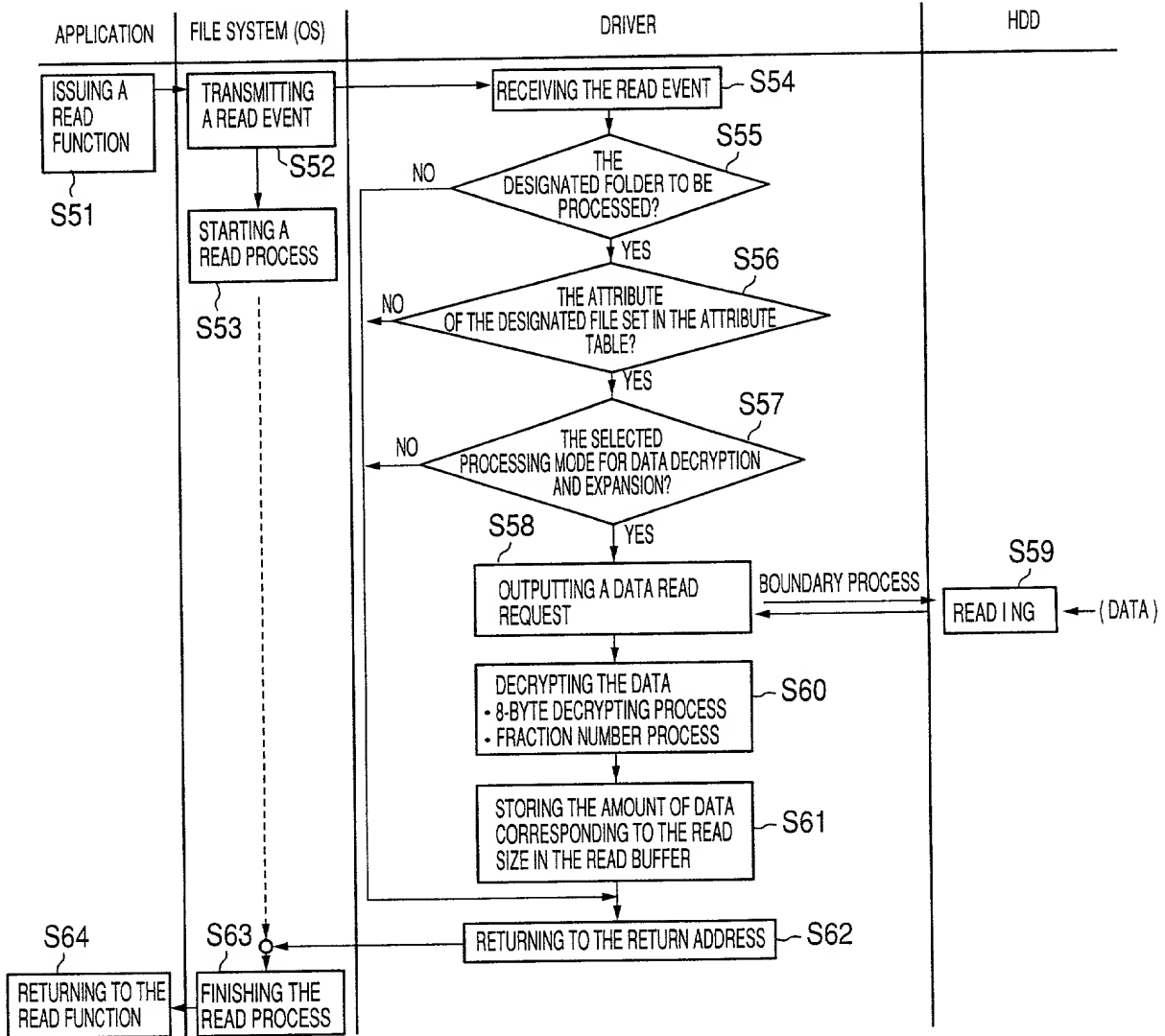
[illegible]

FIG.7

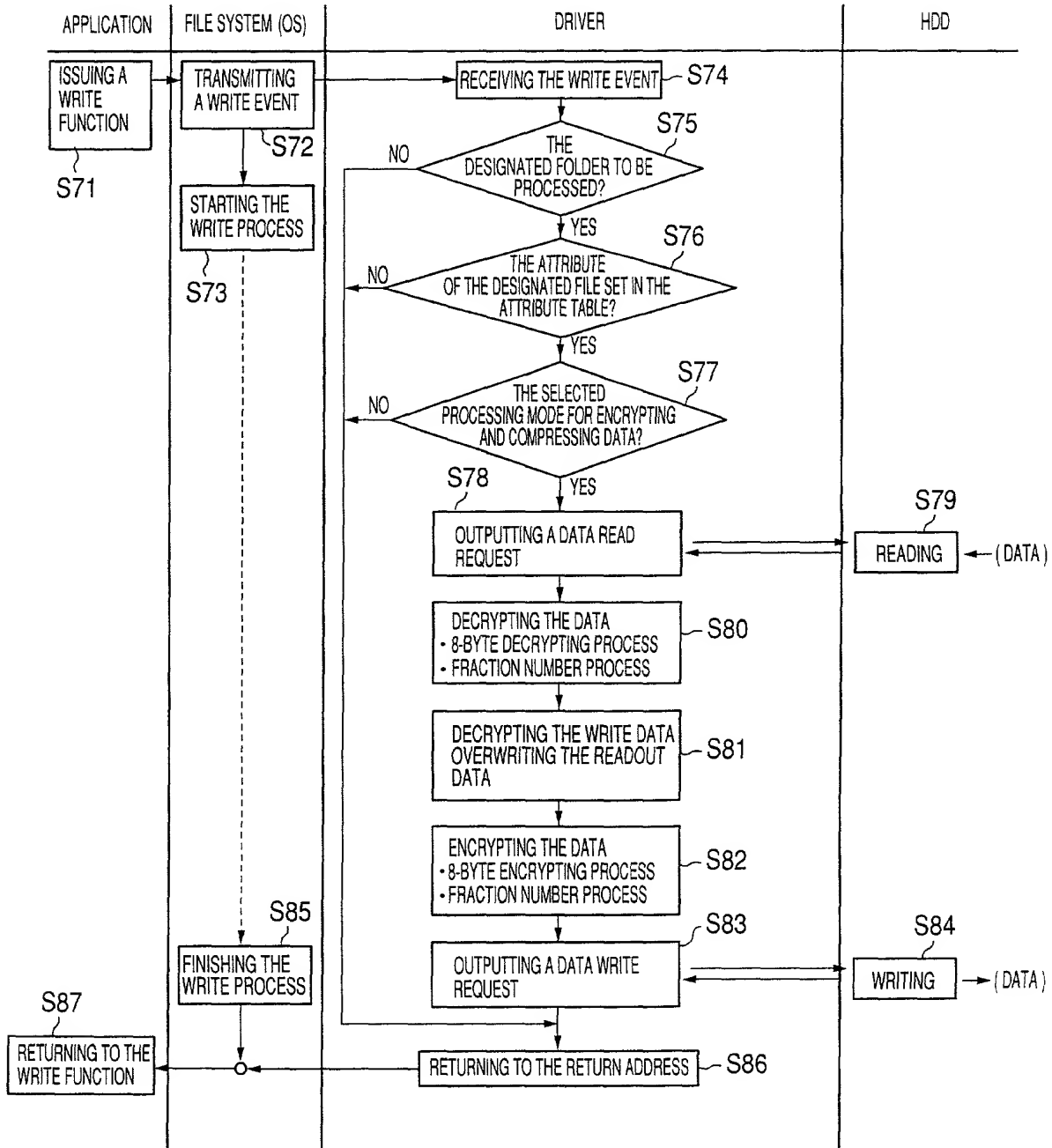
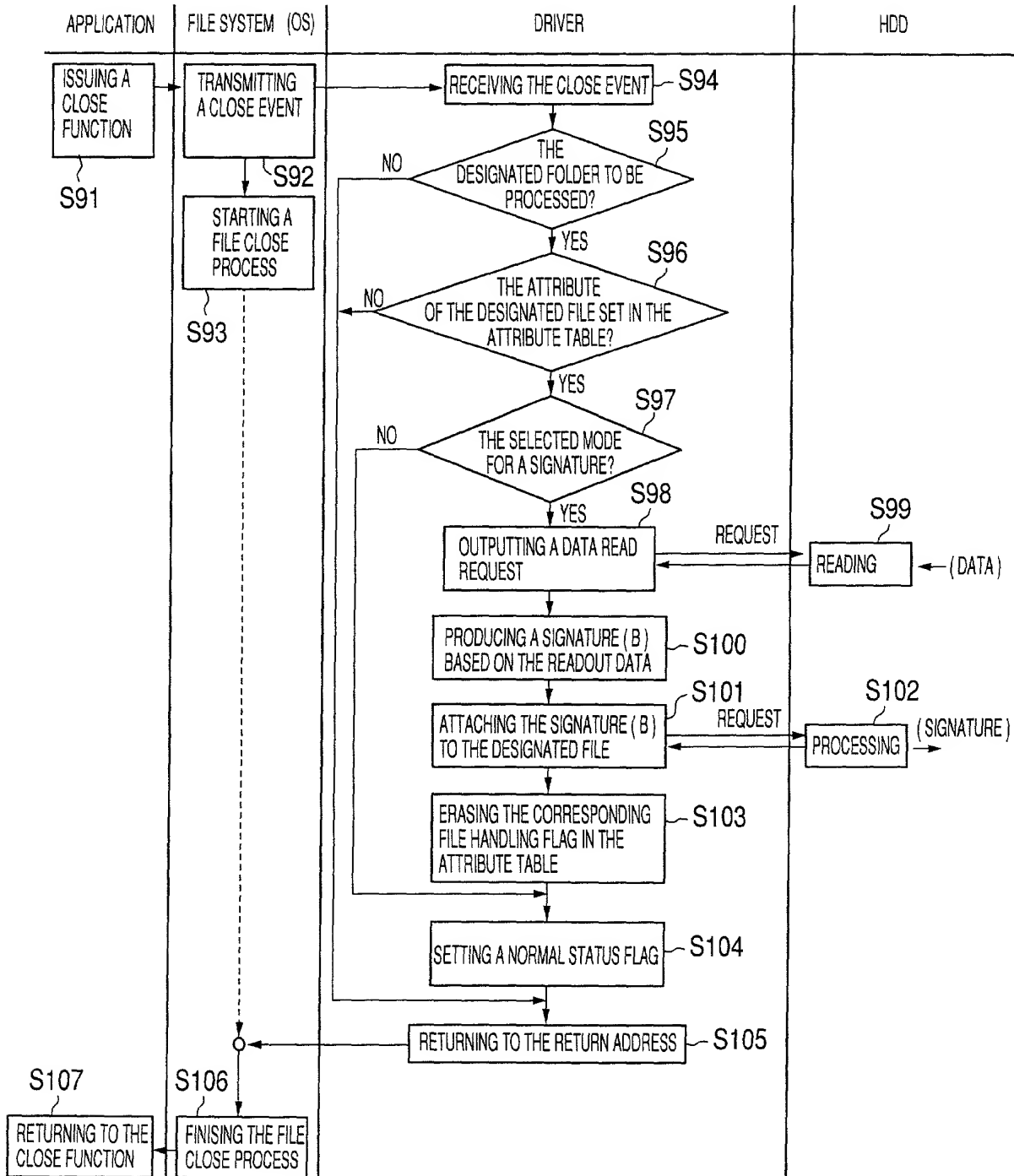


FIG.8



004240" 32165560

FIG.9

OPEN	CLOSE	WRITE	READ	MODE VALUE
SIGNATURE CHECK	SIGNATURE PRODUCTION	ENCRYPTION COMPRESSION	DECRYPTION EXPANSION	_____
x	x	x	x	0
x	x	x	o	1
x	x	o	x	2
x	x	o	o	3
x	o	x	x	4
x	o	x	o	5
x	o	o	x	6
x	o	o	o	7
o	x	x	x	8
o	x	x	o	9
o	x	o	x	10
o	x	o	o	11
o	o	x	x	12
o	o	x	o	13
o	o	o	x	14
o	o	o	o	15

FIG.10

MODE VALUE	FOLDER
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H
8	I
9	J
1 0	K
1 1	L
1 2	M
1 3	N
1 4	O
1 5	P

004240" 004240" 004240"

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Declaration and Power of Attorney For Patent Application

特許出願宣言書及び委任状

Japanese Language Declaration

日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。

My residence, post office address and citizenship are as stated next to my name.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

FILE PROCESSING UNIT

上記発明の明細書（下記の欄でx印がついていない場合は、本書に添付）は、

the specification of which is attached hereto unless the following box is checked:

☐ __月__日に提出され、米国出願番号または特許協定条約国際出願番号を____とし、
（該当する場合）____に訂正されました。

☐ was filed on _____
as United States Application Number or
PCT International Application Number
_____ and was amended on
_____ (if applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第37編第1条56項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Japanese Language Declaration (日本語宣言書)

私は、米国法典第35編119条(a)-(d)項又は365条(b)項に基づき下記の、米国外の国の少なくとも一カ国を指定している特許協力条約365(a)項に基づき国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

Prior Foreign Application(s)

外国での先行出願
Pat. Appln. No. 11-125010 Japan
(Number) (Country)
(番号) (国名)

(Number) (Country)
(番号) (国名)

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Priority Not Claimed

優先権主張なし

30/April/1999
(Day/Month/Year Filed)
(出願年月日)

☐

(Day/Month/Year Filed)
(出願年月日)

☐

私は、第35編米国法典119条(e)項に基づいて下記の米国外特許出願規定に記載された権利をここに主張いたします。

(Application No.) (Filing Date)
(出願番号) (出願日)

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.) (Filing Date)
(出願番号) (出願日)

私は、下記の米国法典第35編120条に基づいて下記の米国外特許出願に記載された権利、又は米国を指定している特許協力条約365条(c)に基づき権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項又は特許協力条約で規定された方法で先行する米国外特許出願に開示されていない限り、その先行米国外出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第37編1条56項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

(Application No.) (Filing Date)
(出願番号) (出願日)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of application.

(Status: Patented, Pending, Abandoned)
(現況: 特許許可済、係属中、放棄済)

(Application No.) (Filing Date)
(出願番号) (出願日)

(Status: Patented, Pending, Abandoned)
(現況: 特許許可済、係属中、放棄済)

私は、私自身の知識に基づいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じることに基づき表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに上記のごとく宣誓を致します。

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration (日本語宣言書)

委任状： 私は下記の発明者として、本出願に関する一切の手続きを米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。(弁理士、または代理人の氏名及び登録番号を明記のこと)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith (list name and registration number)

<u>Attorney</u>	<u>Reg. No.</u>
Patrick G. Burns	29,367
Roger D. Greer	26,174
Lawrence J. Crain	31,497
Steven P. Fallon	35,132

<u>Attorney</u>	<u>Reg. No.</u>
James K. Folker	37,538
Jonathan D. Feuchtwang	41,017
B. Joe Kim	41,895
Joel H. Bootzin	42,343

直接電話連絡先： (名前及び電話番号)

Send Correspondence to:

Direct Telephone Calls to: (name and telephone number)

Patrick G. Burns, Esq.
Greer, Burns & Crain, Ltd.
Sears Tower - Suite 8660
Chicago, IL 60606 (312) 993-0080

唯一または第一発明者名	Full name of sole or first inventor Takaoki Sasaki	
発明者の署名	Inventor's signature <i>Takaoki Sasaki</i>	Date April 21, 2000
住所	Residence Minato-ku, Tokyo, Japan	
国籍	Citizenship Japan	
私書箱	Post Office Address c/o FUJITSU BASIC SOFTWARE CORPORATION, 15-33, Shibaura 4-chome, Minato-ku, Tokyo, 108-8531 Japan	
第二共同発明者	Full name of second joint inventor, if any Takeshi Hatano	
第二共同発明者	Second inventor's signature <i>Takeshi Hatano</i>	Date April 21, 2000
住所	Residence Minato-ku, Tokyo, Japan	
国籍	Citizenship Japan	
私書箱	Post Office Address c/o FUJITSU BASIC SOFTWARE CORPORATION, 15-33, Shibaura 4-chome, Minato-ku, Tokyo, 108-8531 Japan	

(第三以降の共同発明者についても同様に記載し、署名をすること)

(Supply similar information and signature for third and subsequent joint inventors.)

第三共同発明者名		Full name of third joint inventor, if any Ryota Akiyama	
第三共同発明者の署名	日付	Third inventor's signature <i>Ryota Akiyama</i>	Date April 21, 2000
住所		Residence Kawasaki-shi, Kanagawa, Japan	
国籍		Citizenship Japan	
私書箱		Post Office Address c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa, 211-8588 Japan	
第四共同発明者名		Full name of fourth joint inventor, if any Seigo Kotani	
第四共同発明者の署名	日付	Fourth inventor's signature <i>Seigo Kotani</i>	Date April 21, 2000
住所		Residence Kawasaki-shi, Kanagawa, Japan	
国籍		Citizenship Japan	
私書箱		Post Office Address c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa, 211-8588 Japan	
第五共同発明者名		Full name of fifth joint inventor, if any Takayuki Hasebe	
第五共同発明者の署名	日付	Fifth inventor's signature <i>Takayuki Hasebe</i>	Date April 21, 2000
住所		Residence Kawasaki-shi, Kanagawa, Japan	
国籍		Citizenship Japan	
私書箱		Post Office Address c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa, 211-8588 Japan	
第六共同発明者名		Full name of sixth joint inventor, if any	
第六共同発明者の署名	日付	Sixth inventor's signature	Date
住所		Residence	
国籍		Citizenship	
私書箱		Post Office Address	